UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/676,743	09/30/2003	Anandhi Somasekaran	MSFT-2763/305222.1	7938

41505 7590 01/16/2009

WOODCOCK WASHBURN LLP (MICROSOFT CORPORATION)
CIRA CENTRE, 12TH FLOOR
2929 ARCH STREET
PHILADELPHIA, PA 19104-2891

EXAMINER	
WANG, BEN C	

ART UNIT	PAPER NUMBER
2192	

MAIL DATE	DELIVERY MODE
01/16/2009	PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

Office Action Summary	Application No.	Applicant(s)	
	10/676,743	SOMASEKARAN ET AL.	
	Examiner	Art Unit	
	BEN C. WANG	2192	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE *3* MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *22 October 2008*.

2a)☐ This action is **FINAL**. 2b)☒ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1-20,23-35 and 37-65* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1-20, 23-35, and 37-65* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☐ The drawing(s) filed on _____ is/are: a)☐ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☐ Information Disclosure Statement(s) (PTO/SB/08) Paper No(s)/Mail Date _____.
4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____ .
5) ☐ Notice of Informal Patent Application
6) ☐ Other: _____.

DETAILED ACTION

1. Applicant's amendment dated April 28, 2008, responding to the Office action mailed January 28, 2008 provided in the rejection of claims 1-20, 23-35, and 37-65, wherein claim 12 has been amended.

Claims 1-20, 23-35, and 37-65, remain pending in the application and which have been fully considered by the examiner.

Applicant's arguments with respect to claims currently amended have been fully considered but are moot in view of the new grounds of rejection – see *O'Farrel et al.* - art made of record, as applied hereto.

Withdrawal of Finality

2. The interviews conducted on October 9, 2008 and October 22, 2008 and along with the amendment received dated October 22, 2008 were carefully evaluated and the arguments were deemed persuasive. In response, a further search was performed and a new prior art, *O'Farrel* reference, has been added to addresses the claimed "a business process service debugger for remotely debugging a business process service" features.

Claim Rejections – 35 USC § 102(e)

The following is quotation of 35 U.S.C. 102(e) which form the basis for all obviousness rejections set forth in this office action:

> (e) the invention was described in (1) an application for patent, published under section 122(b), by another filed in the United States before the invention by the applicant for patent or (2) a patent granted on an application for patent by another filed in the United States before the invention by the applicant for patent, except that an international application filed under the treaty defined in section 351(a) shall have the effects for purposes of this

subsection of an application filed in the United States only if the international application designated the United States and was published under Article 21(2) of such treaty in the English language.

3. Claims 1-7, 9-11, 40-41, 43-45, 60-61, and 63-65 are rejected under 35 U.S.C. 102(e) as being anticipated by O'Farrel et al. (Pub. No. US 2004/0168155 A1) (hereinafter 'O'Farrel' - art made of record)

4. **As to claim 1** (Previously Presented), O'Farrel discloses a business process service debugger for remotely debugging a business process service (e.g., [0010] - … a debugger allows debugging of applications using flow paradigm …; [0011] - … debugging a software application comprising a plurality of processes that asynchronously co-operate and whose execution is controlled through interaction with run-time code …), comprising:

- means for reading stored state information regarding events related to at least one business process implemented for the business process service (e.g., [0045] - … The hooks ... in a portion of the runtime that corresponds to changes of state in the node of the flow …; [0055] - … in the Java class in the runtime that is associated with the *ActivityState* for activities …);
- means for establishing a communications connection with a remote computer, wherein the remote computer is implementing the business process service (e.g., Fig. 9; [0083] - … flow debugger 64 … to attach to a known one of computing devices … executing run-time code … The computing device may be local to flow debugger 64 or remote, identified by a network address …);

- means for displaying a symbolic representation of the operation of the business process service based on the stored state information (e.g., [0043] - … further receives debug data for presentation by way of a graphical user interface (GUI) which is detailed with reference to Figs. 9-17); and
- means for remotely debugging the one business process service using the symbolic representation, communications connection and stored state information (e.g., Fig. 9; [0083] - … flow debugger 64 … to attach to a known one of computing devices … executing run-time code … The computing device may be local to flow debugger 64 or remote, identified by a network address …)

5. **As to claim 2** (Previously Presented) (incorporating the rejection in claim 1), O'Farrel discloses the business process service debugger, wherein business processes and instances of the business process service other than those being debugged are not disrupted during debugging (e.g., [0011] - … debugging a software application comprising a plurality of processes that asynchronously co-operate and whose execution is controlled through interaction with run-time code …)

6. **As to claim 3** (Previously Presented) (incorporating the rejection in claim 1), O'Farrel discloses the business process service debugger wherein the symbolic representation comprises a workflow of at least one business process in the business process service (e.g., Figs. 12, 13A, 13B, 14)

7. **As to claim 4** (Previously Presented) (incorporating the rejection in claim 1), O'Farrel discloses the business process service debugger further comprising means for interacting with the business process service according to a user instruction (e.g., [0012] - … receiving user input to allow setting of at least one breakpoint in the application using the flow diagram)

8. **As to claim 5** (Original) (incorporating the rejection in claim 1), O'Farrel discloses the business process service debugger wherein the stored state information corresponds to a variable assignment within the business process service (e.g., [0045] - … The hooks ... in a portion of the runtime that corresponds to changes of state in the node of the flow …; [0055] - … in the Java class in the runtime that is associated with the *ActivityState* for activities …)

9. **As to claim 6** (Previously Presented) (incorporating the rejection in claim 1), O'Farrel discloses the business process service debugger wherein the events are historical events that occurred prior to failure of the at least one business process (e.g., [0074] - … debug manager 60 may detect that a fault has occurred in the running code ...)

10. **As to claim 7** (Previously Presented) (incorporating the rejection in claim 1), O'Farrel discloses the business process service debugger wherein the stored state

information corresponds to message flow data and an order in which run time components performed the one business process as a result of message processing (e.g., [0045] - … The hooks ... in a portion of the runtime that corresponds to changes of state in the node of the flow …; [0055] - … in the Java class in the runtime that is associated with the *ActivityState* for activities …)

11. **As to claim 9** (Previously Presented) (incorporating the rejection in claim 1), O'Farrel discloses the business process service debugger wherein the events are events that occur prior to an inserted breakpoint in the one business process (e.g., [0074] - … debug manager 60 may detect that a fault has occurred in the running code ...)

12. **As to claim 10** (Original) (incorporating the rejection in claim 1), O'Farrel discloses the business process service debugger wherein said debugging means comprises means for detecting a location where the instance is being processed (e.g., [0045] - … in a portion of the runtime that corresponds to changes of state in the nodes of the flow …; [0071])

13. **As to claim 11** (Original) (incorporating the rejection in claim 1), O'Farrel discloses the business process service debugger wherein said debugging means comprises means for detecting a location where the instance state is being stored (e.g.,

[0045] - … in a portion of the runtime that corresponds to changes of state in the nodes of the flow …; [0071])

14. **As to claim 40** (Previously Presented), O'Farrel discloses a method in a computer system for displaying on a display device a graphical debugging environment for a business process service (e.g., [0010] - … a debugger allows debugging of applications using flow paradigm …; [0011] - … debugging a software application comprising a plurality of processes that asynchronously co-operate and whose execution is controlled through interaction with run-time code …) , the method comprising:

- obtaining design information about the business process service;
- obtaining tracking information about execution of the business process service (e.g., [0033] - … converts the graphically defined flow, into a plurality tokens, each token representative of a node on the graph. The tokens, in turn, may be stored as entries of a database …; [0034] - Upon execution, flow engine software such as that forming part of run-time code 52 may access the database to extract and interpret the tokens …);
- generating symbolic representation of the operation of the business process service according to the design information and tracking information (e.g.,); and
- displaying on the display device a graphical debugging environment showing the symbolic representation (e.g., [0043] - … further receives debug data for

presentation by way of a graphical user interface (GUI) which is detailed with reference to Figs. 9-17)

15. **As to claim 41** (Original) (incorporating the rejection in claim 40), O'Farrel discloses further comprising receiving runtime data for the business process service and presenting the runtime data on the display device (e.g., [0043] - … further receives debug data for presentation by way of a graphical user interface (GUI) which is detailed with reference to Figs. 9-17)

16. **As to claim 43** (Previously Presented) (incorporating the rejection in claim 40), O'Farrel discloses wherein the graphical debugging environment enables a user to place a breakpoint in the symbolic representation of the operation of the business process service (e.g., [0012] - … receiving user input to allow setting of at least one breakpoint in the application using the flow diagram)

17. **As to claim 44** (Previously Presented) (incorporating the rejection in claim 40), O'Farrel discloses the symbolic representation comprising symbols, wherein the graphical debugging environment also displays information about the symbols (e.g., [0043] - … further receives debug data for presentation by way of a graphical user interface (GUI) which is detailed with reference to Figs. 9-17)

18. **As to claim 45** (Previously Presented) (incorporating the rejection in claim 40), O'Farrel discloses further comprising receiving input from an input device to place a break point proximate a symbol, and presenting a symbol representing the break point on the symbolic representation (e.g., [0012] - … receiving user input to allow setting of at least one breakpoint in the application using the flow diagram; [0043] - … further receives debug data for presentation by way of a graphical user interface (GUI) which is detailed with reference to Figs. 9-17)

19. **As to claim 60** (Previously Presented), O'Farrel discloses a computer-readable storage medium having computer-executable instructions for performing a method for displaying on a display device a graphical debugging environment for a business process service (e.g., [0010] - … a debugger allows debugging of applications using flow paradigm …; [0011] - … debugging a software application comprising a plurality of processes that asynchronously co-operate and whose execution is controlled through interaction with run-time code …), the method comprising:

- obtaining design information about the business process service;
- obtaining configuration information about the business process service (e.g.,) [0033] - … converts the graphically defined flow, into a plurality tokens, each token representative of a node on the graph. The tokens, in turn, may be stored as entries of a database …; [0034] - Upon execution, flow engine software such as that forming part of run-time code 52 may access the database to extract and interpret the tokens …;

- generating symbolic representation of the operation of the business process service according to the design information and configuration information (e.g., [0032] - … IBM WebSphere™ Studio Application Developer Integration Edition (ESAD-IE) engine allows creation of applications in accordance with the flow paradigm, using a graphical user interface ...); and
- displaying on the display device a graphical debugging environment showing the symbolic representation (e.g., [0043] - … further receives debug data for presentation by way of a graphical user interface (GUI) which is detailed with reference to Figs. 9-17)

20. **As to claim 61** (Previously Presented) (incorporating the rejection in claim 60), please refer to claim **41** as set forth accordingly.

21. **As to claims 63-65**, refer to above **claims 43-45**, accordingly.

Claim Rejections – 35 USC § 103(a)

The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

22. Claims 8, 12-20, 23-35, 37-39, 42, 46-59, and 62 are rejected under 35 U.S.C. 103(a) as being unpatentable over O'Farrel in view of Adams et al., (*BizTalk™ Unleashed 1st Edition, Feb. 2002, Sams Publishing*) (hereinafter 'Adams')

23. **As to claim 8** (Original) (incorporating the rejection in claim 1), O'Farrel discloses that a debugger allows debugging of applications using a flow paradigm (e.g., Abstract) but does not explicitly disclose other limitations stated below.

However, in an analogous art of *BizTalk™ Unleashed*, Adams discloses the business process service debugger wherein said reading means further comprises means for reading stored business process service configuration information (e.g., Sec. of "Developing Custom Tracking Solutions" - the two primary tracking activities: configuring tracking settings and viewing tracking data)

Therefore, it would have been obvious to one of ordinary skill in the art, at the time the invention was made to combine the teachings of Adams into the O'Farrel's system to further provide other limitations stated above in the O'Farrel system.

The motivation is that it would further enhance the O'Farrel's system by taking, advancing and/or incorporating Adams's system which offers significant advantages for building customized tracking solutions using the infrastructure provided by Biztalk server as once suggested by Adams (e.g., Sec. of "Developing Custom Tracking Solution")

24. **As to claim 12** (Currently amended), O'Farrel discloses a system for remotely debugging a distributed transactional application, comprising:

- a server configured to execute an instance of a business process service, thereby generating runtime data (e.g., [0011] - … debugging a software application comprising a plurality of processes that asynchronously co-operate and whose execution is controlled through interaction with run-time code …)
- a client computer configured to execute a debugging user interface (UI) process that establishes a communications connection with the server requests runtime data for at least one of the plurality of business processes (e.g., Fig. 9; [0083] - … flow debugger 64 … to attach to a known one of computing devices … executing run-time code … The computing device may be local to flow debugger 64 or remote, identified by a network address …), and generates, based on the runtime data, a symbolic representation of the business service process showing any debugging break points specified by a user (e.g., [0043] - … further receives debug data for presentation by way of a graphical user interface (GUI) which is detailed with reference to Figs. 9-17);

Further, O'Farrel discloses that a debugger allows debugging of applications using a flow paradigm (e.g., Abstract) but does not explicitly disclose other limitations stated below.

However, in an analogous art of *BizTalk™ Unleashed*, Adams discloses an interceptor for monitoring the runtime data and, when a specified break point is identified, causing the server to enter or leave a debugging state (e.g., Sec. of "Developing Custom Tracking Solution" – to perform the two primary tracking activities:

configuring tracking settings and viewing tracking data; building customized tracking solutions using the infrastructure provided by BizTalk™ Server; Sec. of "Example: A Custom Tracking Solution", 4th Par. – this submits an interchange to BizTalk™ Server and returns a submission identifier; the submission ID should be displayed in the result window)

Therefore, it would have been obvious to one of ordinary skill in the art, at the time the invention was made to combine the teachings of Adams into the O'Farrel's system to further provide other limitations stated above in the O'Farrel system.

The motivation is that it would further enhance the O'Farrel's system by taking, advancing and/or incorporating Adams's system which offers significant advantages for building customized tracking solutions using the infrastructure provided by Biztalk server as once suggested by Adams (e.g., Sec. of "Developing Custom Tracking Solution")

25. **As to claim 13** (Original) (incorporating the rejection in claim 12), O'Farrel discloses the system further comprising for storing business process service state information (e.g., [0045] - … The hooks ... in a portion of the runtime that corresponds to changes of state in the node of the flow …; [0055] - … in the Java class in the runtime that is associated with the *ActivityState* for activities …)

Further, Adams discloses a database for receiving the runtime data (e.g., Sec. of "Developing Custom tracking Solutions" – we begin by covering the tracking database, including the core tables where interchange data, document data, and routing data is stored)

26. **As to claim 14** (Previously Presented) (incorporating the rejection in claim 13), O'Farrel discloses the system further comprising a display device, for displaying the symbolic representation (e.g., [0043] - … further receives debug data for presentation by way of a graphical user interface (GUI) which is detailed with reference to Figs. 9-17) and a user input device, wherein the input device is used to specify debugging break points (e.g., [0012] - … receiving user input to allow setting of at least one breakpoint in the application using the flow diagram)

27. **As to claim 15** (Previously Presented) (incorporating the rejection in claim 14), O'Farrel discloses the system wherein the symbolic representation comprises presents a workflow representative of the program flow of the business process service (e.g., [0043] - … further receives debug data for presentation by way of a graphical user interface (GUI) which is detailed with reference to Figs. 9-17)

28. **As to claim 16** (Previously Presented) (incorporating the rejection in claim 14), O'Farrel discloses the system wherein the display device further displays data representative of a message flow of the business process service (e.g., [0043] - … further receives debug data for presentation by way of a graphical user interface (GUI) which is detailed with reference to Figs. 9-17)

29. **As to claim 17** (Previously Presented) (incorporating the rejection in claim 14), O'Farrel discloses the system wherein the symbolic representation is presented

according to stored state information (e.g., [0045] - … The hooks ... in a portion of the runtime that corresponds to changes of state in the node of the flow …; [0055] - … in the Java class in the runtime that is associated with the *ActivityState* for activities …; [0045] - … The hooks ... in a portion of the runtime that corresponds to changes of state in the node of the flow …; [0055] - … in the Java class in the runtime that is associated with the *ActivityState* for activities …)

30. **As to claim 18** (Previously Presented) (incorporating the rejection in claim 12), Adams discloses the system wherein a message box database is coupled between the server and client computer and is configured for communicating debugging requests from the client computer (e.g., Sec. of "BizTalk™ Messaging" – BizTalk™ Messaging is a set of services that provide a mechanism for routing documents in an enterprise environment; Sec. of "Configuration Objects Versus BizTalk™ Messaging Objects" – Biztalk™ Messaging objects are the actual objects stored in the Biztalk™ Messaging database)

31. **As to claim 19** (Previously Presented) (incorporating the rejection in claim 18), Adams discloses the system wherein the UI process comprises an application program interface for communicating with the message box database (e.g., Sec. of "BizTalk™ Messaging" – BizTalk™ Messaging is a set of services that provide a mechanism for routing documents in an enterprise environment; Sec. of "Configuration Objects Versus

BizTalk™ Messaging Objects" – Biztalk™ Messaging objects are the actual objects stored in the Biztalk™ Messaging database)

32. **As to claim 20** (Previously Presented) (incorporating the rejection in claim 18), Adams discloses the system further comprising a tracking database to receive business process service tracking information (e.g., Sec. of "Developing Custom tracking Solutions" – we begin by covering the tracking database, including the core tables where interchange data, document data, and routing data is stored), and further O'Farrel discloses wherein the UI process comprises a UI component for communicating with the tracking database (e.g., [0033] - … converts the graphically defined flow, into a plurality tokens, each token representative of a node on the graph. The tokens, in turn, may be stored as entries of a database …; [0034] - Upon execution, flow engine software such as that forming part of run-time code 52 may access the database to extract and interpret the tokens …)

33. **As to claim 23** (Original) (incorporating the rejection in claim 12), Adams discloses the system wherein the interceptor is a component of a computer language that provides stored state tracking information (e.g., Sec. of "Developing Custom Tracking Solution" – to perform the two primary tracking activities: configuring tracking settings and viewing tracking data; building customized tracking solutions using the infrastructure provided by BizTalk™ Server; Sec. of "Example: A Custom Tracking

Solution", 4th Par. – this submits an interchange to BizTalk™ Server and returns a submission identifier; the submission ID should be displayed in the result window)

34. **As to claim 24** (Original) (incorporating the rejection in claim 12), O'Farrel discloses the system wherein the UI process detects a location where the instance is being processed (e.g., [0045] - … in a portion of the runtime that corresponds to changes of state in the nodes of the flow …; [0071])

35. **As to claim 25** (Original) (incorporating the rejection in claim 12), O'Farrel discloses the system wherein the UI process detects a location where the instance state is being stored (e.g., [0045] - … in a portion of the runtime that corresponds to changes of state in the nodes of the flow …; [0071])

36. **As to claim 26** (Previously Presented), O'Farrel discloses a method for debugging an instance of a business process service running on a remote computer (e.g., [0010] - … a debugger allows debugging of applications using flow paradigm …; [0011] - … debugging a software application comprising a plurality of processes that asynchronously co-operate and whose execution is controlled through interaction with run-time code …), comprising:

- generating for display, in a graphical user interface (GUI), a symbolic representation of the business process service based on a correlation of information about the design and execution of the business process service (e.g.,

[0043] - … further receives debug data for presentation by way of a graphical user interface (GUI) which is detailed with reference to Figs. 9-17);

- receiving a debugging command generated by a user interacting with the GUI (e.g., [0012] - … receiving user input to allow setting of at least one breakpoint in the application using the flow diagram);
- a direct client connection channel with the remote compute (e.g., Fig. 9; [0083] - … flow debugger 64 … to attach to a known one of computing devices … executing run-time code … The computing device may be local to flow debugger 64 or remote, identified by a network address …);
- receiving a runtime request, generated by a user interacting with the GUI, for event information generated by execution of the instance of the business process service (e.g., [0012] - … receiving user input to allow setting of at least one breakpoint in the application using the flow diagram);
- sending the runtime request to the remote computer for processing by the remote computer (e.g., Fig. 9; [0083] - … flow debugger 64 … to attach to a known one of computing devices … executing run-time code … The computing device may be local to flow debugger 64 or remote, identified by a network address …)

Further, O'Farrel discloses that a debugger allows debugging of applications using a flow paradigm (e.g., Abstract) but does not explicitly disclose other limitations stated below.

However, in an analogous art of *BizTalk™ Unleashed*, Adams discloses causing an interceptor to monitor runtime data generated by the instance of the business

process in accordance with the debugging command (e.g., Sec. of "Developing Custom Tracking Solution" – to perform the two primary tracking activities: configuring tracking settings and viewing tracking data; building customized tracking solutions using the infrastructure provided by BizTalk™ Server; Sec. of "Example: A Custom Tracking Solution", 4th Par. – this submits an interchange to BizTalk™ Server and returns a submission identifier; the submission ID should be displayed in the result window)

Therefore, it would have been obvious to one of ordinary skill in the art, at the time the invention was made to combine the teachings of Adams into the O'Farrel's system to further provide other limitations stated above in the Charisius system.

The motivation is that it would further enhance the Charisius' system by taking, advancing and/or incorporating Adams's system which offers significant advantages for building customized tracking solutions using the infrastructure provided by Biztalk server as once suggested by Adams (e.g., Sec. of "Developing Custom Tracking Solution")

37. **As to claim 27** (Original) (incorporating the rejection in claim 26), Adams discloses further comprising: querying a database containing a status of the business process service; displaying a query result on a display device; receiving user input with respect to the query result (e.g., Sec. of "Tracking Events in the Tracking Database", 1st Par. – the BizTalk™ Document Tracking Web application all you to query the database for document flows between organizations or applications; the XLANG™ events captured in the database can be viewed in relationship with the message flows displayed by document tracking Web application)

Further, O'Farrel discloses establishing the direct client connection channel in response to the user input (e.g., Fig. 9; [0083] - … flow debugger 64 … to attach to a known one of computing devices … executing run-time code … The computing device may be local to flow debugger 64 or remote, identified by a network address …)

38. **As to claim 28** (Original) (incorporating the rejection in claim 27), O'Farrel discloses the information contained in the database is instance runtime data (e.g., [0033] - … converts the graphically defined flow, into a plurality tokens, each token representative of a node on the graph. The tokens, in turn, may be stored as entries of a database …; [0034] - Upon execution, flow engine software such as that forming part of run-time code 52 may access the database to extract and interpret the tokens …)

39. **As to claim 29** (Original) (incorporating the rejection in claim 27), Adams discloses the information contained in the database is instance tracking data (e.g., Sec. of "Developing Custom Tracking Solution" – to perform the two primary tracking activities: configuring tracking settings and viewing tracking data; building customized tracking solutions using the infrastructure provided by BizTalk™ Server; Sec. of "Example: A Custom Tracking Solution", 4th Par. – this submits an interchange to BizTalk™ Server and returns a submission identifier; the submission ID should be displayed in the result window)

40. **As to claim 30** (Previously Presented) (incorporating the rejection in claim 26), O'Farrel discloses further comprising: creating the business process service using a process designer (e.g., [0032] - … IBM WebSphere™ Studio Application Developer Integration Edition (ESAD-IE) engine allows creation of applications in accordance with the flow paradigm, using a graphical user interface ...)

Further, Adams discloses saving a business process service configuration and symbolic data in a database as information about the design of the business process service (e.g., Sec. of "Developing Custom Tracking Solution" – to perform the two primary tracking activities: configuring tracking settings and viewing tracking data; building customized tracking solutions using the infrastructure provided by BizTalk™ Server; Sec. of "Example: A Custom Tracking Solution", 4th Par. – this submits an interchange to BizTalk™ Server and returns a submission identifier; the submission ID should be displayed in the result window); Furthermore, O'Farrel discloses displaying the symbolic representation on a display device according to the saved business process service configuration and symbolic data; generating a runtime request based on the symbolic representation; and displaying a result of the runtime request on the display device (e.g., [0043] - … further receives debug data for presentation by way of a graphical user interface (GUI) which is detailed with reference to Figs. 9-17)

41. **As to claim 31** (Previously Presented) (incorporating the rejection in claim 30), O'Farrel discloses the symbolic representation comprises a shape corresponding to an operation in the business process service (e.g., [0032] - … IBM WebSphere™ Studio

Application Developer Integration Edition (ESAD-IE) engine allows creation of applications in accordance with the flow paradigm, using a graphical user interface ...)

42. **As to claim 32** (Previously Presented) (incorporating the rejection in claim 30), O'Farrel discloses the symbolic representation comprises a workflow representation of the business process service (e.g., [0032] - … IBM WebSphere™ Studio Application Developer Integration Edition (ESAD-IE) engine allows creation of applications in accordance with the flow paradigm, using a graphical user interface ...)

43. **As to claim 33** (Original) (incorporating the rejection in claim 30, O'Farrel discloses the saving step takes place in connection with compiling and deploying the business process service (e.g., [0032] - … IBM WebSphere™ Studio Application Developer Integration Edition (ESAD-IE) engine allows creation of applications in accordance with the flow paradigm, using a graphical user interface ...)

44. **As to claim 34** (Original) (incorporating the rejection in claim 30), O'Farrel discloses the business process service is implemented in a computer language that provides stored state information (e.g., [0032] - … IBM WebSphere™ Studio Application Developer Integration Edition (ESAD-IE) engine allows creation of applications in accordance with the flow paradigm, using a graphical user interface ...)

45. **As to claim 35** (Previously Presented) (incorporating the rejection in claim 26), O'Farrel discloses the debugging command is a break point (e.g., [0012] - … receiving user input to allow setting of at least one breakpoint in the application using the flow diagram)

46. **As to claim 37** (Previously Presented) (incorporating the rejection in claim 26), O'Farrel discloses the runtime data is state information (e.g., [0045] - … The hooks ... in a portion of the runtime that corresponds to changes of state in the node of the flow …; [0055] - … in the Java class in the runtime that is associated with the *ActivityState* for activities …)

47. **As to claim 38** (Original) (incorporating the rejection in claim 26), O'Farrel discloses further comprising detecting a location where the instance is being processed (e.g., [0045] - … in a portion of the runtime that corresponds to changes of state in the nodes of the flow …; [0071])

48. **As to claim 39** (Original) (incorporating the rejection in claim 26), O'Farrel discloses further comprising detecting a location where an instance state is being stored (e.g., [0045] - … in a portion of the runtime that corresponds to changes of state in the nodes of the flow …; [0071])

49. **As to claim 42** (Previously Presented) (incorporating the rejection in claim 41), Adams discloses wherein the runtime data comprises historical message flow information including identification of run time messages that were constructed as a result of processing received messages, and further comprises order information pertaining to the order in which different run time components were executed as a result of processing received messages (e.g., Chapter 10 – Using the BizTalk Orchestration Designer, 2nd Par. – the BizTalk Orchestration Designer contains wizards that guide you through creating and configuring ports and message flows; Sec. of "Example: A Custom Tracking Solution", 4th Par. – this submits an interchange to BizTalk™ Server and returns a submission identifier; the submission ID should be displayed in the result window)

50. **As to claim 46** (Previously Presented), O'Farrel discloses a computer-readable storage medium having computer-executable instructions for performing a method for debugging an instance of a business process service running on a remote computer (e.g., [0010] - … a debugger allows debugging of applications using flow paradigm …; [0011] - … debugging a software application comprising a plurality of processes that asynchronously co-operate and whose execution is controlled through interaction with run-time code …), comprising:

- generating for display, in a graphical user interface (GUI), a symbolic representation of the business process service based on a correlation of information about the design and execution of the business process services

(e.g., [0043] - … further receives debug data for presentation by way of a graphical user interface (GUI) which is detailed with reference to Figs. 9-17);

- receiving a debugging command generated by a user interacting with the GUI (e.g., [0012] - … receiving user input to allow setting of at least one breakpoint in the application using the flow diagram);
- establishing a direct client connection channel with the remote computer (e.g., Fig. 9; [0083] - … flow debugger 64 … to attach to a known one of computing devices … executing run-time code … The computing device may be local to flow debugger 64 or remote, identified by a network address …);
- receiving a runtime request; and sending the runtime request to the remote computer for processing by the remote computer (e.g., Fig. 9; [0083] - … flow debugger 64 … to attach to a known one of computing devices … executing run-time code … The computing device may be local to flow debugger 64 or remote, identified by a network address …)

Further, O'Farrel discloses that a debugger allows debugging of applications using a flow paradigm (e.g., Abstract) but does not explicitly disclose other limitations stated below.

However, in an analogous art of *BizTalk™ Unleashed*, Adams discloses causing an interceptor to monitor runtime data generated by the instance of the business process service in accordance with the debugging command (e.g., Sec. of "Developing Custom Tracking Solution" – to perform the two primary tracking activities: configuring tracking settings and viewing tracking data; building customized tracking solutions using

the infrastructure provided by BizTalk™ Server; Sec. of "Example: A Custom Tracking Solution", 4th Par. – this submits an interchange to BizTalk™ Server and returns a submission identifier; the submission ID should be displayed in the result window)

Therefore, it would have been obvious to one of ordinary skill in the art, at the time the invention was made to combine the teachings of Adams into the O'Farrel's system to further provide other limitations stated above in the O'Farrel system.

The motivation is that it would further enhance the O'Farrel's system by taking, advancing and/or incorporating Adams's system which offers significant advantages for building customized tracking solutions using the infrastructure provided by Biztalk server as once suggested by Adams (e.g., Sec. of "Developing Custom Tracking Solution")

51. **As to claims 47-55**, refer to above **claims 27-35**, accordingly.

52. **As to claim 56** (Previously Presented) (incorporating the rejection in claim 30), O'Farrel discloses the computer-readable storage medium, wherein the debugging command is a request for data regarding an instance of the business process service (e.g., [0043] - … provides debug commands to the debug manger …)

53. **As to claims 57-59**, refer to above **claims 37-39**, accordingly.

54. **As to claim 62**, refer to above **claims 42**, accordingly.

Conclusion

55. Any inquiry concerning this communication or earlier communications from the examiner should be directed to Ben C. Wang whose telephone number is 571-270-1240. The examiner can normally be reached on Monday - Friday, 8:00 a.m. - 5:00 p.m., EST.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Tuan Q. Dam can be reached on 571-272-3695. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

/Ben C Wang/
Ben C. Wang
Examiner, Art Unit 2192

/Tuan Q. Dam/
Supervisory Patent Examiner, Art Unit 2192